Homework 4: Graph Spectra

Submission Date: 2016-12-18

In this homework you will implement spectral graph clustering as described in the paper "On Spectral Clustring: Analysis and an algorith" (http://ai.stanford.edu/~ang/papers/nips01-spectral.pdf). You can choose any language to implement the task. Using Matlab/Octave the algorithm can be implemented using few lines of code. As a KTH student you can obtain a student license for Matlab (https://www.kth.se/student/kth-it-support/software/download/matlab).

It is recommended that you read the above mentioned paper first. As implementing the algorithm should be trivial. As a warm up task, you should first compute the eigenvectors and eigenvalues of a set of graphs, and find out how many communities do these graphs have. You will find it trivial if you are already familiar with Matlab.

Matlab Tutorial

All the graphs that you will use for this homework are comma separated edge lists.

Importing comma separated edge list in Matlab

```
>> E = csvread('/..path../file.dat')
```

Converting Edge list to adjacency matrix

```
>> col1 = E(:,1);
>> col2 = E(:,2);
>> max_ids = max(max(col1,col2));
>> As= sparse(col1, col2, 1, max_ids, max_ids);
>> A = full(As)
```

Getting the eigenvalues

```
>> [v,D] = eig(A)
```

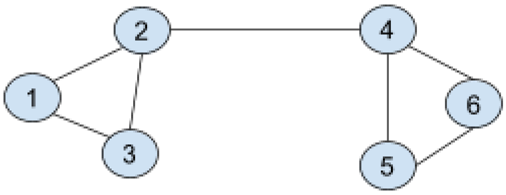
Sort eigenvalues

```
>> sort(diag(D))
```

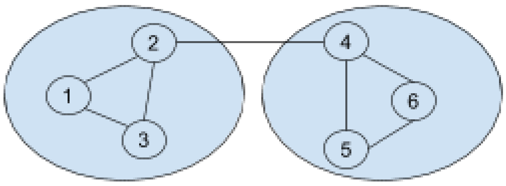
Spectral Graphs Partitioning

Fiedler Vector: The eigenvector corresponding to second smallest eigenvalue of Laplacian matrix, *L,* is called Fiedler Vector. If the graph has two modules then it bisects the graph into only two communities based on the sign of the corresponding vector entry.

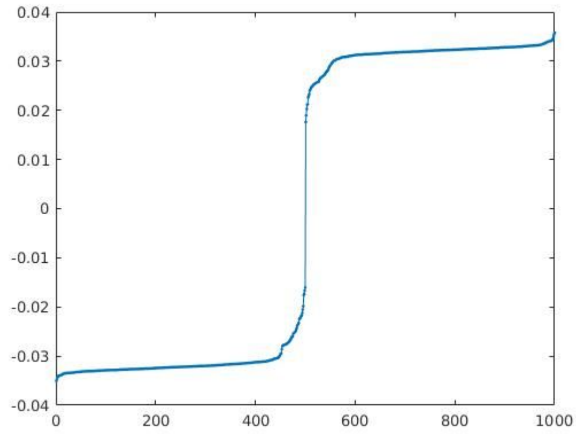
Example:



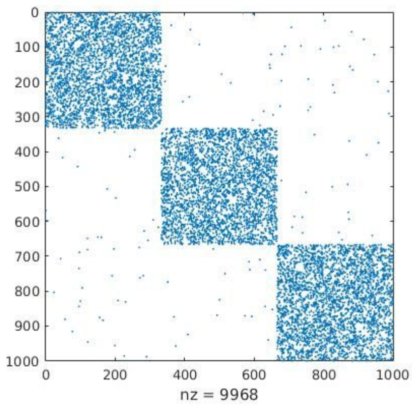
Node #	Eigenvectors					
1	0.1928	-0.3511	0	0.4934	-0.7071	0
2	0.3214	-0.3049	-0.5345	-0.7574	0.7071	-0.0546
3	0.2571	-0.5035	0.5345	0.1147	0	0.0546
4	0.5143	0.1524	-0.5345	0.3787	0	-0.0546
5	0.5143	0.5035	0.2673	-0.1147	0	-0.6761
6	0.5143	0.5035	0.2673	-0.1147	0	0.7307
Eigenvalues	0	0.6972	3	4.3028	4	3



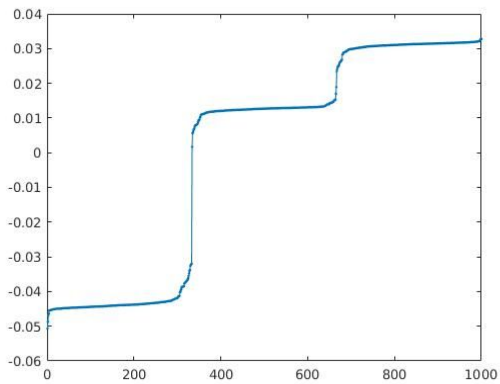
If we plot the sorted Fiedler Vector it the communities in the graphs will be clearly visible. For example for a graph with 1000 nodes and two communities the graph for sorted Fiedler vector might look like this



What if there are more than two communities.



Sparsity Pattern



Sorted Fiedler Vector

For very large graphs use eigs function to Find to find few eigenvalues and eigenvectors of a matrix, i.e.,

```
>> [V D] = eigs(L, 2, 'SA');
```

What if there are nested clusters?

Two basic approaches

1) **Recursive partitioning (Hagen et al.,'91)**
    - Recursively apply partitioning algorithm in a hierarchical divisive manner.
    - Disadvantages: Inefficient, unstable
2) **Cluster Multiple Eigenvector, i.e., K-eigenvector Algorithm (Ng et al.,'01)**
    http://ai.stanford.edu/~ang/papers/nips01-spectral.pdf

Exercise

Your task is to implement K-eigenvector Algorithm as described in by Ng et al., 01. Using your implementation of K-eigenvector you will analyze the graphs that are provided with the exercise.

- Real graph **"example1.dat"**

This data set was prepared by Ron Burt. He dug out the 1966 data collected by Coleman, Katz and Menzel on medical innovation. They had collected data from physicians in four towns in Illinois, Peoria, Bloomington, Quincy and Galesburg.
http://moreno.ss.uci.edu/data.html#ckm

- Synthetic graph **"example2.dat"**

What to deliver:

- Matlab source code
- 1 - 2 page report on the analysis of the given graphs